

Ice Sheet System model Software Architecture

Eric LAROUR¹, Eric RIGNOT^{1,3}, Mathieu MORLIGHEM^{1,2}, Hélène SEROUSSI^{1,2} Chris BORSTAD¹, Feras HABBAL^{1,3}, Daria HALKIDES^{1,4}, Behnaz KHAKBAZ¹, John SCHIERMEIER¹, Nicole SCHLEGEL¹

¹Jet Propulsion Laboratory - California Institute of Technology

²Laboratoire MSSMat, École Centrale Paris, France

³University of California, Irvine

⁴Joint Institute for Regional Earth System Science & Engineering, UCLA



[Software Architecture](#)[Larour et al.](#)[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

Outline

① Introduction

② General Layout

③ C/C++ Code

④ MATLAB

⑤ Conclusions

[Software Architecture](#)

Larour et al.

Introduction

[Introduction](#)

General Layout

C/C++ Code

MATLAB

Conclusions

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

[Software Architecture](#)[Larour et al.](#)

Introduction

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

[Software Architecture](#)[Larour et al.](#)

Introduction

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

[Software Architecture](#)[Larour et al.](#)

Introduction

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

- ISSM is written in C++ (C-style)
- ISSM is hosted in MATLAB using the mex API
- ISSM is written to run serial/parallel seamlessly
- ISSM is configured to run using the autotools suite for multi-platform support

Software Architecture

Larour et al.

Introduction

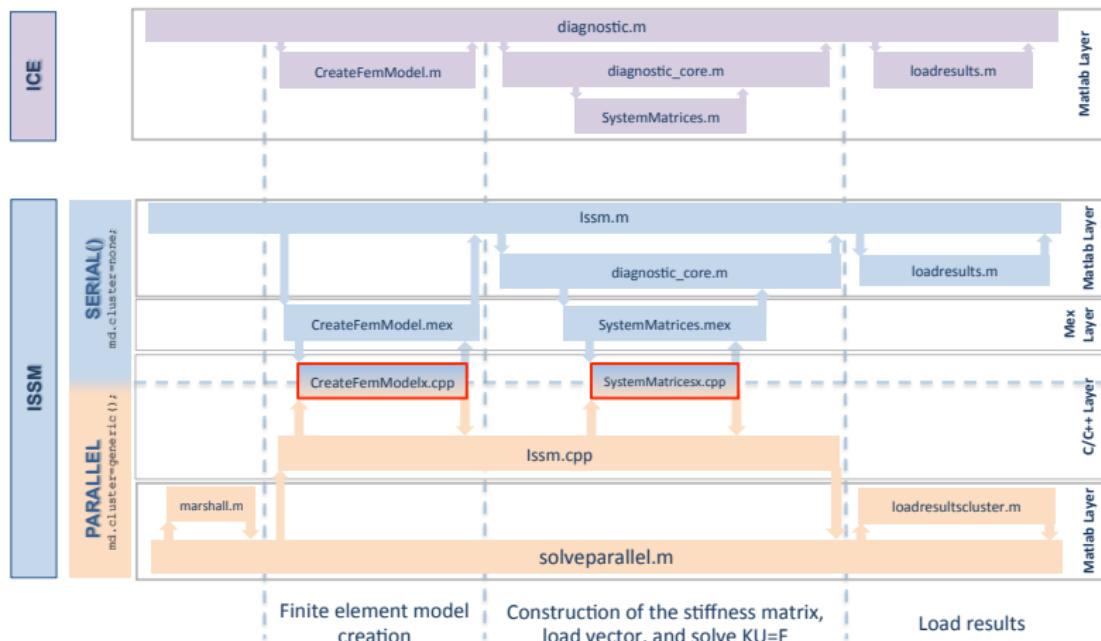
General Layout

C/C++ Code

MATLAB

Conclusions

General Layout



[Software Architecture](#)[Larour et al.](#)[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

General Layout

```
$ pwd  
/Users/cborstad/issmuci/trunk  
$ ls  
AUTHORS      NEWS          configs       examples      missing  
COPYING      README        configure.ac execution    scripts  
ChangeLog    bin           cron          externalpackages src  
INSTALL      compile       depcomp      install-sh   startup.m  
LICENSE      config.guess doc           lib          startup.py  
Makefile.am  config.sub   etc           m4           test
```

[Software Architecture](#)

Larour et al.

[Introduction](#)

[General Layout](#)

[C/C++ Code](#)

[MATLAB](#)

[Conclusions](#)

General Layout

The core of the code is in trunk/src, for C++, mex API, MATLAB routines, etc.

```
$ pwd  
/Users/cborstad/issmuci/trunk/src  
$ ls  
Makefile.am ad c dox m mex perl pro py
```

The core of the C/C++ code is in trunk/src/c

```
$ pwd  
/Users/cborstad/issmuci/trunk/src/c  
$ ls  
Container include issm.h shared toolkits  
EnumDefinitions intel-compile.sh modules solutions  
Makefile.am io objects solvers
```

[Software Architecture](#)

Larour et al.

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

C/C++ Code

Solutions are high-level drivers of C/C++ routines:

```
$ pwd  
/Users/cborstad/issmuci/trunk/src/c/solutions  
$ ls  
AdjointCorePointerFromSolutionEnum.cpp diagnostic_core.cpp  
AnalysisConfiguration.cpp enthalpy_core.cpp  
CorePointerFromSolutionEnum.cpp gradient_core.cpp  
ProcessArguments.cpp hydrology_core.cpp  
ResetBoundaryConditions.cpp hydrology_core_step.cpp  
WriteLockFile.cpp issm.cpp  
adjointbalancethickness_core.cpp objectivefunctionC.cpp  
adjointdiagnostic_core.cpp prognostic_core.cpp  
balancethickness_core.cpp solutions.h  
bedslope_core.cpp steadystate_core.cpp  
control_core.cpp steadystateconvergence.cpp  
controlconvergence.cpp surfaceslope_core.cpp  
controlrestart.cpp thermal_core.cpp  
controltao_core.cpp thermal_core_step.cpp  
convergence.cpp transient_core.cpp
```

[Software Architecture](#)[Larour et al.](#)[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

C/C++ Code

Solutions call other modules or solutions. Femmodel is driving class.

```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/solutions
$ cat surfaceslope_core.cpp
/*!\file: surfaceslope_core.cpp

...
#include "./solutions.h"
...
void surfaceslope_core(FemModel* femmodel) {
/*parameters: */
int dim;
...
/*Recover some parameters: */
femmodel->parameters->FindParam(&dim,MeshDimensionEnum);
...
/*Call on core computations: */
...
solver_linear(femmodel);
...
if(solution_type==SurfaceSlopeSolutionEnum && !control_analysis){
InputToResultx(femmodel->elements,femmodel->nodes,femmodel->vertices,
femmodel->loads,femmodel->materials,femmodel->parameters, SurfaceSlopeXEnum
}
...
}
```

[Software Architecture](#)

Larour et al.

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

C/C++ Code

Extensive list of modules that drive the computations:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/modules
$ ls
AddExternalResultx      InputUpdateFromConstantx      OutputRiftsx
AverageFilterx          InputUpdateFromDakotax       ParsePetscOptionsx
AverageOntoPartitionx   InputUpdateFromMatrixDakotax PointCloudFindNeighborsx
BamgConvertMeshx        InputUpdateFromSolutionx    PropagateFlagsFromConnectivityx
BamgTriangulatex       InputUpdateFromVectorDakotax Reduceloadx
Bamgx                  InputUpdateFromVectorx       Reducevectorgtofx
Chacoxx                InterpFromGridToMeshx       Reducevectorgtosx
ComputeBasalStressx    InterpFromMesh2dx         RequestedOutputsx
ComputeStrainRatex     InterpFromMeshToGridx      ResetConstraintsx
ConfigureObjectsx       InterpFromMeshToMesh2dx    ResetCoordinateSystemx
ConstraintsStatex      InterpFromMeshToMesh3dx    Responsex
ContourToMeshx          IoModelToConstraintsx     RheologyBbarAbsGradientx
ContourToNodesx         KMLMeshWritex           Scotchx
ControlInputGetGradientx KMLOverlayx            Shp2Kmlx
ControlInputScaleGradientx Kml2Expx              Solverx
ControlInputSetGradientx Ll2xyx                SpcNodesx
CostFunctionx          MassFluxx              StringToEnumx
CreateNodalConstraintsx MaxAbsVxx             SurfaceAbsVelMisfitx
DakotaResponsesx       MaxAbsVy              SurfaceAreax
Dakotax                MaxAbsVzx             SurfaceAverageVelMisfitx
DragCoefficientAbsGradientx MaxVelx               SurfaceLogVelMisfitx
ElementConnectivityx    MaxVxx                SurfaceLogVxVyMisfitx
ElementResponsex       MaxVy                 SurfaceRelVelMisfitx
EnumToStringx          MaxVzx                SystemMatricesx
Exp2Kmlx               Mergesolutionfromftogx ThicknessAbsGradientx
GetSolutionFromInputsx MeshPartitionx         ThicknessAbsMisfitx
GetVectorFromInputsx   MeshProfileIntersectionx TimeAdaptx
Gradjx                 MinVelx              TriaSearchx
GroundinglineMigrationx MinVxx               UpdateConstraintsx
HoleFillerx            MinVy                UpdateDynamicConstraintsx
IceVolumex             MinVzx               UpdateVertexPositionsx
InputArtificialNoisex  ModelProcessorx        VecMergex
InputControlUpdatex    NodalValuex          VerticesDofx
InputConvergencex      NodeConnectivityx      Xy2llx
InputDuplicatex         NodesDofx            modules.h
InputScalex             Orthx
```

[Software Architecture](#)

Larour et al.

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

C/C++ Code

Typical module to create stiffness and loads:

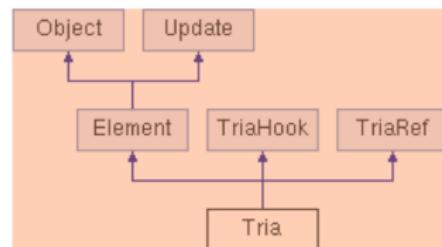
```
$ pwd
/Users/cborstad/issmuci/trunk/src/c/modules/SystemMatricesx
$ ls
SystemMatricesx.cpp  SystemMatricesx.h

...
if(kflag){

Kff=NewMat(fsize,fsize,connectivity,numberofdofspernode);
Kfs=NewMat(fsize,ssize,connectivity,numberofdofspernode);
df=NewVec(fsize);

/*Fill stiffness matrix from elements: */
for (i=0;i<elements->Size();i++){
element=(Element*)elements->GetObjectByOffset(i);
element->CreateKMatrix(Kff,Kfs,df);
}

/*Fill stiffness matrix from loads if loads have the current
for (i=0;i<loads->Size();i++){
load=(Load*)loads->GetObjectByOffset(i);
if (load->InAnalysis(configuration_type)) load->CreateKMatrix(Kff,Kfs);
}
...
}
```



[Software Architecture](#)

Larour et al.

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

MATLAB

MATLAB routines for pre-post processing:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/m
$ ls
classes enum kml model planet qmu shared solutions solvers utils
$ pwd
/Users/cborstad/issmuci/trunk/src/m/model
$ ls
BasinConstrain.m           graddetection.m          process_solve_options.m
BasinConstrainShelf.m      ismodelselfconsistent.m processgeometry.m
DepthAverage.m              kmlimagesc.m            project2d.m
MeltingGroundingLines.m    loadmultipleresultsfromcluster.m project3d.m
PropagateFlagsUntilDistance.m loadresultsfromcluster.m qpr.m
README                     loadresultsfromdisk.m   qstat.m
SectionValues.m             marshall.m              radarpower.m
ThicknessCorrection.m       mechanicalproperties.m recover_areas.m
WriteData.m                 mesh                     regionaltransient2d.m
addnote.m                  misfit.m                setflowequation.m
averageconnectivity.m       modelextract.m         setmask.m
averaging.m                 modelextract.m         setmask2.m
basalstress.m               modelextractfromdomains.m shear2d.m
basevert.m                 modis.m                 sia.m
bedslope.m                 multiplequeue          slope.m
collapse.m                 outflow.m               solve.m
contourenvelope.m          parametercontroloptimization.m solveparallel.m
contourmassbalance.m        parameterization        solvers
display                    parseresultsfromdisk.m thicknessvolution.m
divergence.m                partition               tres.m
drivingstress.m            petscversion.m         waitonlock.m
effectivepressure.m         plot
extrude.m                  printmodel.m
```

[Software Architecture](#)

Larour et al.

[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

MATLAB

Augmented by mex routine:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/mex
$ ls
AddExternalResult      InputConvergence      ParsePetscOptions
AverageFilter          InputDuplicate        PointCloudFindNeighbors
BamgConvertMesh        InputScale            ProcessParams
BamgMesher             InputToResult        PropagateFlagsFromConnectivity
BamgTriangulate        InputUpdateFromConstant ReduceLoad
Chaco                  InputUpdateFromDakota Reducevectororgof
ComputeBasalStress     InputUpdateFromSolution Reducevectorgtos
ConfigureObjects       InputUpdateFromVector ResetCoordinateSystem
ConstraintsState       InternalFront        Response
ContourToMesh          InterpFromGridToMesh Scotch
ContourToNodes         InterpFromMesh2d    Shp2Kml
ControlInputGetGradient InterpFromMeshToGrid Solver
ControlInputScaleGradient InterpFromMeshToMesh2d SparseToVector
ControlInputSetGradient InterpFromMeshToMesh3d SpcNodes
ControlOptimization    KMLMeshWrite        StringToEnum
CostFunction           KMLOverlay          SurfaceArea
CreateNodalConstraints Km12Exp            SystemMatrices
Dakota                 L12xy              Test
DakotaResponses        Makefile.am        TimeAdapt
Echo                  Mergesolutionfromftog TriMesh
ElementConnectivity    MeshPartition       TriMeshNoDensity
EnumToString           MeshProfileIntersection TriMeshProcessRifts
Exp2Kml               ModelProcessor      TriMeshRefine
GetSolutionFromInputs  NodeConnectivity    TriaSearch
GetVectorFromInputs    NodesDof            UpdateConstraints
Gradj                 Orth                UpdateDynamicConstraints
GroundinglineMigration OutputResults       UpdateVertexPositions
HoleFiller             OutputRiffts       VerticesDof
InputControlUpdate     ParameterOutput    Xy211
```

[Software Architecture](#)[Larour et al.](#)[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

MATLAB

Typical mex routine:

```
$ pwd
/Users/cborstad/issmuci/trunk/src/mex/SystemMatrices
$ ls
SystemMatrices.cpp  SystemMatrices.h
void mexFunction( int nlhs, mxArray* plhs[], int nrhs, const mxArray* prhs[]){

    /*input datasets: */
    Elements    *elements    = NULL;
    ...
    /* output datasets: */
    Mat        Kff    = NULL;
    ...
    /*Boot module: */
    MODULEBOOT();
    ...
    /*Input datasets: */
    FetchMatlabData((DataSet**)&elements,ELEMENTS);
    ...
    SystemMatricesx(&Kff,&Kfs,&pf,&df,&kmax,elements,nodes,vertices,loads,materials,parameters,kflag,pflag);
    ...
    /*write output datasets: */
    WriteMatlabData(KFF,Kff);
    ...
    /*end module: */
    MODULEEND();
```

[Software Architecture](#)[Larour et al.](#)[Introduction](#)[General Layout](#)[C/C++ Code](#)[MATLAB](#)[Conclusions](#)

Conclusions

- ISSM is at its core a C++ language, wrapped into Matlab
- ISSM can run both in serial and parallel mode, either by relying on mex modules calling the C-code, or on issm.exe compiled out of the entire C++ code
- More features are available, such as:
 - Nightly runs and test suites
 - Autotools to automatically install on any platform
 - Automatic documentation

A wide-angle photograph of a desolate, icy terrain. In the foreground, a flat expanse of white, textured snow or ice stretches across the frame. Beyond it, a range of mountains rises, their peaks covered in thick, white snow. The mountains are rugged, with deep shadows in the valleys and bright reflections on the snow. The sky above is a clear, pale blue, with a few wispy clouds near the horizon.

Thanks!